BLUEBIKES MANAGEMENT SYSTEM

DAMG 6210: DATABASE MANAGEMENT & DATABASE DESIGN

GROUP 7

Name	NUID
Nirmayee Dighe	002984202
Yash Nema	001562522
Sushmita Maity	001092534
Mohit Nagpal	001502719
Arjun Janardhan	001566405
Mahendra Nanuram Kaswan	002195119

Table of Contents

1. OVERVIEW

Bluebikes ride sharing is one of the highlights of the bicycle transportation in Boston Metropolitan area. With more than 393 bike stations and fleet of 3800 bikes, it is going very much hand-in-hand with Boston's reputation of being one of the most walkable cities. And, since the onset of Covid pandemic, Bluebikes has become an essential mode of commute for students, working class people as well as travelers. Bluebikes are easily available and affordable promoting self-distancing during pandemic.

As of 2021, close to 14million rides on Bluebikes have been completed and it is projected to grow exponentially with refreshed knack for fitness amongst public in the wake of pandemic. Users can easily rent out the Bluebike using the Bluebike app or from Bluebike station kiosk. There are also multiple kinds of passes available depending on the duration of membership as well as pass value, with inclusion of income and non-income based subsidies for inclusion of all sects of society.

But, along with growing popularity and increased usage, it has become tough to conclude how efficient management of Bluebikes is. In line to keep the Bluebikes management system thriving, it is essential to choose an efficient Relational Database Management System (RDBMS) so that students won't miss on their important classes, working class people are not being late to their work, least loss of man-hours in traffic everyday as well as not keeping the riders waiting for their rides. A relational database management system will address the user issues, security concerns and update problems significantly, if it is designed in a way that it is capable of handling these issues.

2. PROBLEM STATEMENT

A Bluebike management database system will manage all areas of the application including customer registration, wallet setup, employee, and customer service functions. With its flexibility, power and easy-to-use properties, it is designed to meet needs of customers and employees both.

3. OBJECTIVES

1. Boosting the application with better database design for a better user experience, ensuring higher user retention.
2. To develop an automated system that processes customer information, bike rides, payments, and customer service quickly, securely and patiently.
3. To store customer information, wallets, bike information, docking station availability, employee workflow.
4. To minimize the bike ride charges
5. To provide top tier bike renting services and better coordination between different departments.

4. PROPOSED SOLUTION

Solution is broken down into three areas:

1. **Booking management** - Setting up account, take a subscription, book a bike, call a customer care, making a payment
2. **Customer Care management** – Assign/ Pick tickets from queue.
3. **Employee management** – Employee details log

Booking Management

- Allowing the user to connect with the nearest Bluebike station by using the Bike/ rental/docking Table and Customer Table.

- Book the bike seamlessly using the app where they will get details from the Rent table (payment id)
- Calculate the time of ride using Rental table(difference between Drop time and start time and multiply with multiplicative factor consisting of coupon code, membership type and fixed rate/hour)
- Easy Cost calculation using the rental, discount and membership table.

Customer Care

- Ensure fast customer service by assigning an agent immediately using the Customer, Rental, Dock, Ticketing queue and Employee table

Employee Management

- Tracking employee in company and check if they belong to customer care service or who are their managers using Employee table

5. ASSUMPTIONS AND FACTS

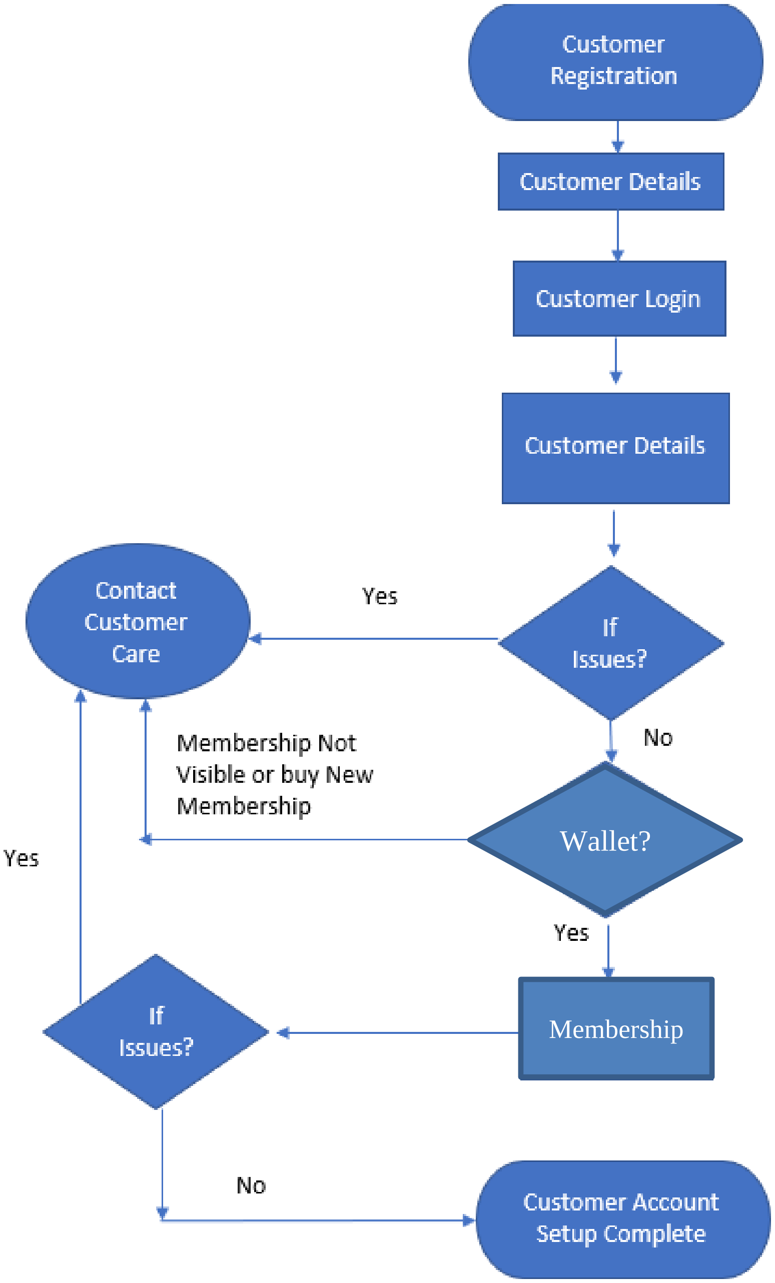
- One customer can book only one bike at a time.
- Customers will have to return the bike before booking a new bike.
- A single discount coupon is used during payment.
- A dock can have multiple bikes at any given time. A dock can have a minimum of zero bikes when all bikes have been taken by customers.
- Membership and discounts can be null. It is possible that a customer booking a bike is not a member and is not using any discount while booking a bike.
- Membership id is not unique to each customer.
- We will have a fixed number of coupons available.
- All employees including customer care along with reporting managers are known as employees Bluebikes.
- There are only three membership plans: Weekly, Monthly, and Yearly Pass
- Discounts apply to customers who can be members or non-members
- The customer returns the bike after the ride at the dock and completes the necessary payment
- Each payment is calculated using the time difference between drop and pick up time and multiplicative factors like rate/hour, discount, and membership factor
- All Bluebike are currently functioning only in Boston USA

6. BUSINESS RULES

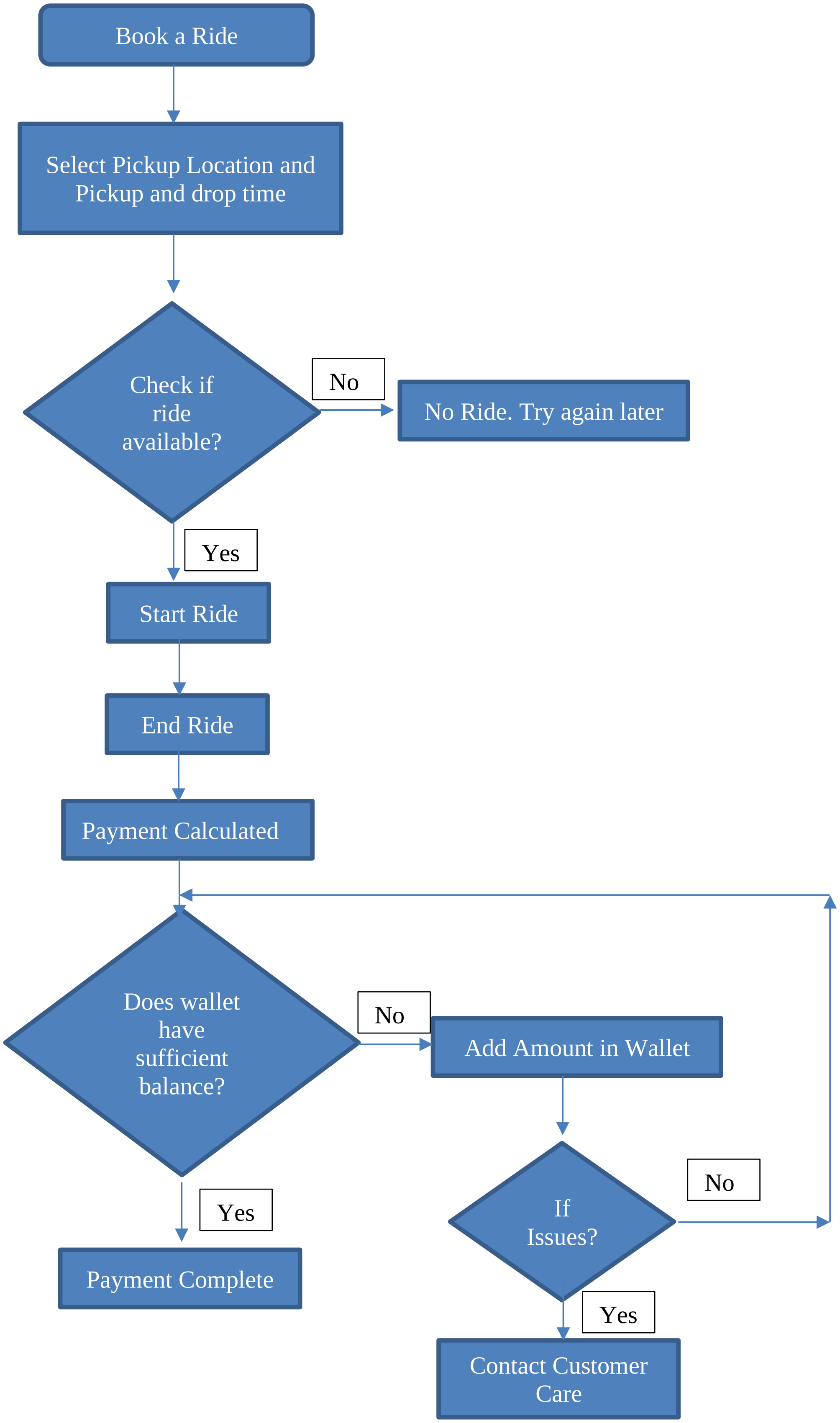
- All Blue bikes are currently **functioning only in Boston USA.**
- **One customer** can book only **one bike** at a time.
- Customers will have to return the bike before booking a new bike.
- Customer should possess an app with all his details (especially registered phone number) and wallet setup on the app
- A customer can only a book a ride for certain hours
- Customer cannot book a ride without the amount not being present in the wallet
- The customer returns the bike after the ride at the same dock station and completes the necessary payment
- There are only three membership plans: **Weekly, Monthly, and Yearly Pass**
- Discounts apply to customers who can be members or non-members
- **Membership** and **discounts** can be **null**. It is possible that a customer booking a bike is not a member and is not using any discount while booking a bike.
- **Membership id** is **not unique** to each customer.
- We will have a fixed number of coupons available
- A **single discount** coupon is used during payment per ride
- There can be **only one membership code** for a **customer**
- A dock can have multiple bikes at any given time. A dock can have a minimum of zero bikes when all bikes have been taken by customers.
- All employees including customer care along with reporting managers are known as employees Bluebikes.
- Each **payment** is calculated using the time **difference between drop and pick up** time and multiplicative factors like **rate/hour, discount, and membership factor**
- **Only one wallet ID** per customer
- The calculated amount is deducted from the wallet of the customer
- Multiple Ticket ID’s can be assigned to a single customer
- Employee Status by default is set to available
- Coupon status can be Active or Inactive
- Bike Status can be available or unavailable, and only the available bikes can be booked.

7. FLOW DIAGRAM

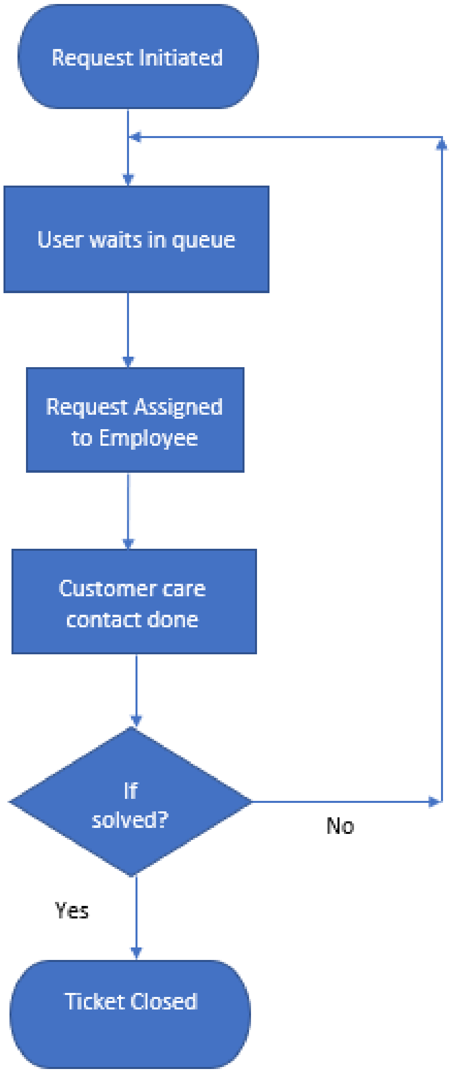
Customer Registration Process



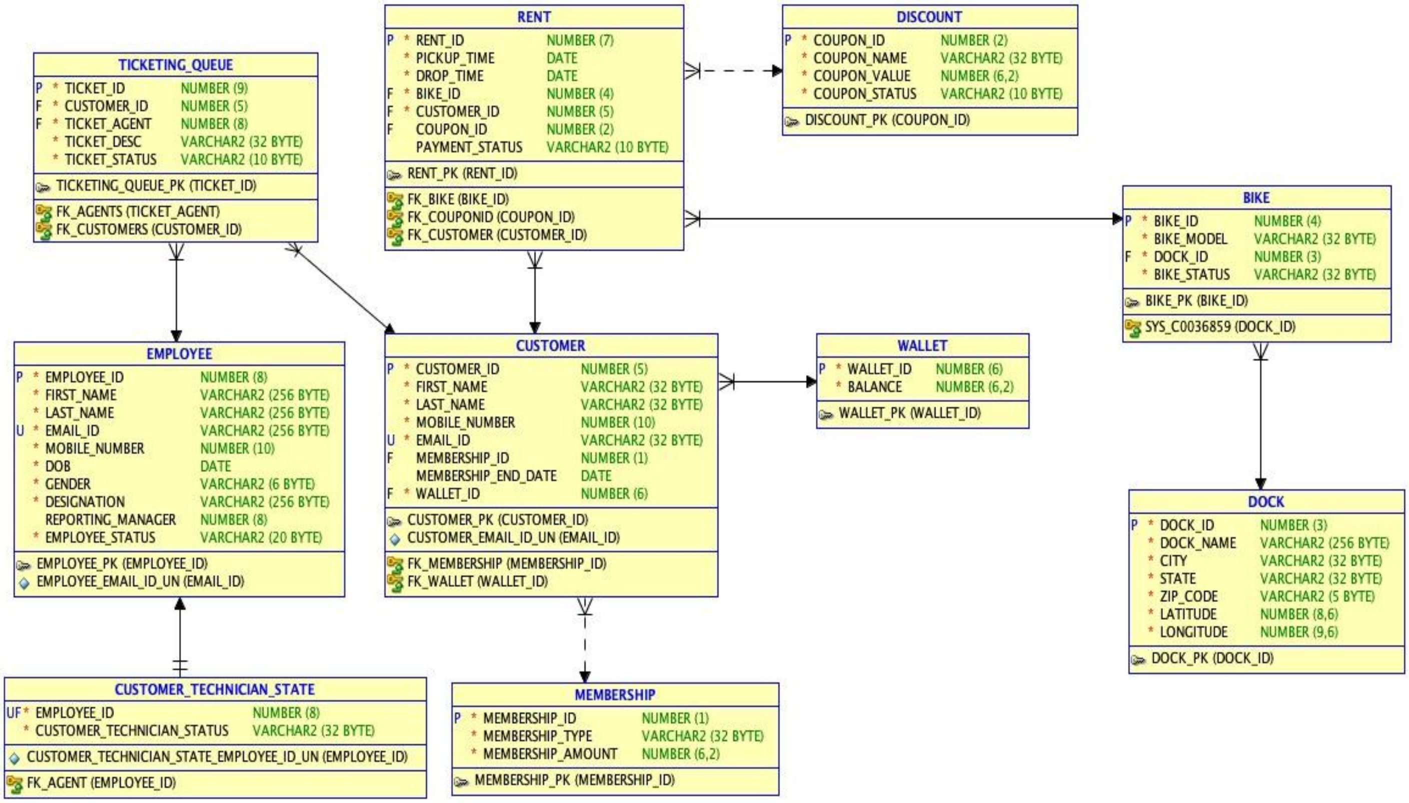
Ride Booking Process



Ticket Handling Process



8. ENTITY RELATIONSHIP DIAGRAM



```
RENT
P  * RENT_ID          NUMBER (7)
   * PICKUP_TIME      DATE
   * DROP_TIME        DATE
F  * BIKE_ID          NUMBER (4)
F  * CUSTOMER_ID      NUMBER (5)
F    COUPON_ID        NUMBER (2)
     PAYMENT_STATUS   VARCHAR2 (10 BYTE)
RENT_PK (RENT_ID)
FK_BIKE (BIKE_ID)
FK_COUPONID (COUPON_ID)
FK_CUSTOMER (CUSTOMER_ID)
```

```
DISCOUNT
P  * COUPON_ID        NUMBER (2)
   * COUPON_NAME      VARCHAR2 (32 BYTE)
   * COUPON_VALUE     NUMBER (6,2)
   * COUPON_STATUS    VARCHAR2 (10 BYTE)
DISCOUNT_PK (COUPON_ID)
```

```
TICKETING_QUEUE
P  * TICKET_ID        NUMBER (9)
F  * CUSTOMER_ID      NUMBER (5)
F  * TICKET_AGENT     NUMBER (8)
   * TICKET_DESC      VARCHAR2 (32 BYTE)
   * TICKET_STATUS    VARCHAR2 (10 BYTE)
TICKETING_QUEUE_PK (TICKET_ID)
FK_AGENTS (TICKET_AGENT)
FK_CUSTOMERS (CUSTOMER_ID)
```

```
BIKE
P  * BIKE_ID          NUMBER (4)
   * BIKE_MODEL       VARCHAR2 (32 BYTE)
F  * DOCK_ID          NUMBER (3)
   * BIKE_STATUS      VARCHAR2 (32 BYTE)
BIKE_PK (BIKE_ID)
SYS_C0036859 (DOCK_ID)
```

```
EMPLOYEE
P  * EMPLOYEE_ID         NUMBER (8)
   * FIRST_NAME          VARCHAR2 (256 BYTE)
   * LAST_NAME           VARCHAR2 (256 BYTE)
U  * EMAIL_ID            VARCHAR2 (256 BYTE)
   * MOBILE_NUMBER       NUMBER (10)
   * DOB                 DATE
   * GENDER              VARCHAR2 (6 BYTE)
   * DESIGNATION         VARCHAR2 (256 BYTE)
     REPORTING_MANAGER   NUMBER (8)
   * EMPLOYEE_STATUS     VARCHAR2 (20 BYTE)
EMPLOYEE_PK (EMPLOYEE_ID)
EMPLOYEE_EMAIL_ID_UN (EMAIL_ID)
```

```
CUSTOMER
P  * CUSTOMER_ID           NUMBER (5)
   * FIRST_NAME            VARCHAR2 (32 BYTE)
   * LAST_NAME             VARCHAR2 (32 BYTE)
   * MOBILE_NUMBER         NUMBER (10)
U  * EMAIL_ID              VARCHAR2 (32 BYTE)
F    MEMBERSHIP_ID         NUMBER (1)
     MEMBERSHIP_END_DATE   DATE
F  * WALLET_ID             NUMBER (6)
CUSTOMER_PK (CUSTOMER_ID)
CUSTOMER_EMAIL_ID_UN (EMAIL_ID)
FK_MEMBERSHIP (MEMBERSHIP_ID)
FK_WALLET (WALLET_ID)
```

```
WALLET
P  * WALLET_ID        NUMBER (6)
   * BALANCE          NUMBER (6,2)
WALLET_PK (WALLET_ID)
```

```
DOCK
P  * DOCK_ID          NUMBER (3)
   * DOCK_NAME        VARCHAR2 (256 BYTE)
   * CITY             VARCHAR2 (32 BYTE)
   * STATE            VARCHAR2 (32 BYTE)
   * ZIP_CODE         VARCHAR2 (5 BYTE)
   * LATITUDE         NUMBER (8,6)
   * LONGITUDE        NUMBER (9,6)
DOCK_PK (DOCK_ID)
```

```
CUSTOMER_TECHNICIAN_STATE
UF * EMPLOYEE_ID                  NUMBER (8)
   * CUSTOMER_TECHNICIAN_STATUS   VARCHAR2 (32 BYTE)
CUSTOMER_TECHNICIAN_STATE_EMPLOYEE_ID_UN (EMPLOYEE_ID)
FK_AGENT (EMPLOYEE_ID)
```

```
MEMBERSHIP
P  * MEMBERSHIP_ID       NUMBER (1)
   * MEMBERSHIP_TYPE     VARCHAR2 (32 BYTE)
   * MEMBERSHIP_AMOUNT   NUMBER (6,2)
MEMBERSHIP_PK (MEMBERSHIP_ID)
```

9. ENTITY TABLES

CUSTOMER

Attributes	Data Type	Comments	Description
CUSTOMER_ID	NUMBER(5)	PRIMARY KEY	A unique ID to identify each customer
FIRST_NAME	VARCHAR(32 BYTE)	NOT NULL	Customer First name
LAST_NAME	VARCHAR(32 BYTE)	NOT NULL	Customer Last Name
MOBILE_NUMBER	NUMBER(10)	NOT NULL	Customer Mobile Number
EMAIL_ID	VARCHAR2(32 BYTE)	NOT NULL	Customer Email ID
MEMBERSHIP_ID	NUMBER(1)	FOREIGN KEY	Customer Membership ID if he possesses any membership
WALLET_ID	NUMBER(6)	FOREIGN KEY	Customer Wallet ID
MEMBERSHIP_END_DATE	DATE	CAN BE NULL	The Membership End date if any

MEMBERSHIP

Attributes	Data Type	Comments	Description
MEMBERSHIP_ID	NUMBER(1)	PRIMARY KEY	A unique ID to identify the kind of membership
MEMBERSHIP_TYPE	VARCHAR2(32 BYTE)	NOT NULL	Membership description eg- weekly, monthly, yearly
MEMBERSHIP_AMOUNT	NUMBER(6,2)	NOT NULL	Membership Amount Paid by the Customer

BIKE

Attributes	Data Type	Comments	Description
BIKE_ID	NUMBER(4)	PRIMARY KEY	A unique ID to identify a bike
BIKE_MODEL	VARCHAR2(32 BYTE)	NOT NULL	Model/Design of the bike
BIKE_STATUS	VARCHAR2(32 BYTE)	NOT NULL	Describes whether the bike is available or in use
DOCK_ID	NUMBER(3)	FOREIGN KEY	The Dock ID where the bike is docked

DOCK

Attributes	Data Type	Comments	Description
DOCK_ID	NUMBER(3)	PRIMARY KEY	A unique ID to identify the different docks around the city
DOCK_NAME	VARCHAR2(256 BYTE)	NOT NULL	Dock Name where the bikes are docked
ZIP_CODE	NUMBER2(5 BYTE)	NOT NULL	Zip Code of the location where the doc is present
STATE	VARCHAR2(32 BYTE)	NOT NULL	State where the bike operates
LATITUDE	NUMBER(8,6)	NOT NULL	Latitude of the bike
LONGITUDE	NUMBER(9,6)	NOT NULL	Longitudinal Location of the bike
CITY	VARCHAR2(32 BYTE)	NOT NULL	City where the bike operates

RENT

Attributes	Data Type	Comments	Description
PICKUP_TIME	TIMESTAMP	NOT NULL	The time when a bike was activated/undocked for a ride
DROP_TIME	TIMESTAMP	NOT NULL	The time when a bike was deactivated/docked after a ride
PAYMENT_ID	NUMBER(7)	NOT NULL	The unique ID used to complete the payment
BIKE_ID	NUMBER(4)	FOREIGN KEY	The bike ID of the bike which was used for the ride
CUSTOMER_ID	NUMBER(5)	FOREIGN KEY	The ID of the Customer using the bike for a ride
COUPON_ID	NUMBER(2)	FOREIGN KEY	ID of any coupon used during the ride
PAYMENT_STATUS	VARCHAR2(10 BYTE)		Payment status of the payment

DISCOUNT

Attributes	Data Type	Comments	Description
COUPON_ID	NUMBER(2)	PRIMARY KEY	ID of any coupon available
COUPON_NAME	VARCHAR2(32 BYTE)	NOT NULL	Name of any coupon used during the ride
COUPON_VALUE	NUMBER(6,2)	NOT NULL	Value of the coupon used during the ride
COUPON_STATUS	VARCHAR2(10 BYTE)	NOT NULL	Status of Coupons

EMPLOYEE

Attributes	Data Type	Comments	Description
EMPLOYEE_ID	NUMBER(8)	PRIMARY KEY	Unique ID for each employee
FIRST_NAME	VARCHAR2(256 BYTE)	NOT NULL	Employee First Name
LAST_NAME	VARCHAR2(256 BYTE)	NOT NULL	Employee Last Name
DOB	DATE	NOT NULL	Employee Date Of Birth
GENDER	VARCHAR2(6 BYTE)	NOT NULL	Employee Gender
DESIGNATION	VARCHAR2(256 BYTE)	NOT NULL	Employee Role/ Designation
REPORTING_MANAGER	NUMBER(8)	NOT NULL	Reporting Manager of the Employee
MOBILE_NUMBER	NUMBER(10)	NOT NULL	Employee Contact Number
EMAIL _ID	VARCHAR2(256 BYTE)	NOT NULL	Employee Email ID
EMPLOYEE_STATUS	VARCHAR2(20 BYTE)	NOT NULL	Employee status in the organization

CUSTOMER_TECHNICIAN_STATE

Attributes	Data Type	Comments	Description
EMPLOYEE_ID	NUMBER(8)	UNIQUE AND FOREIGN KEY	A unique ID to identify a bike
CUSTOMER_TECHNICIAN_STATUS	VARCHAR2(32 BYTE)	NOT NULL	Customer Technician Status

10. SECURITY RULES

5 Roles have been created with the respective Table access.

	Table Names								
Role	Customer	Employee	Ticketing	Dock	Wallet	Rent	Membership	Discount	Bike
User	W			R	R		R	R	R
Customer Care	R, W, U		R, U		R, U	R, U	R, W, U	R, W, U	R, W, U
Employee	R		R, U		W	R			
Manager	R	R, W	R, U	R, W	R, W, U				
Admin	R, W, U	R, W, U	R, W, U	R, W, U	R, W, U	R, W, U	R, W, U	R, W, U	R, W, U

Note: R- Read, W- Write, U- Update, blanks indicate no access

VIEWS TABLE

For the 5 Roles we have described below the views the roles will be able to see.

Role Names	View access (tables)	Description
User	Customer, Wallet, Membership, Discount	User will have access to his own details only
Admin	Rent, Discount, Wallet, Bike, Dock, Customer, Membership, Ticketing_Queue, Employee	Admin will have access to all customer details, ticketing queue and employee details
Customer Care	Ticketing_Queue, Customer, Wallet, Rent, Bike, Membership, Discount	Customer Care will have access to customer details, bike details and membership details
Manager	Employee, Ticketing_Queue, Customer, Dock, Wallet	Manager will have access to customer details, employee details, wallet details, ticketing queue
Employee	Wallet, Ticketing_Queue, Customer	Employee will have access to customer details, wallet details and ticketing queue

PROJECT 4 – DAMG 6210

Steps to follow for script execution:

1) Setting Environment (One time execution): Run File 1 Script. This creates admin for the system ‘nagpalm’. This user will have access like admin. This user creates and drops all tables, sequences and is responsible for inserting data in the respective tables.
2) Drop and Create tables, sequences, procedures (Multiple times execution): Using the ‘nagpalm’ credentials (username: ‘nagpalm’ , password: ‘DMDDuser2#123’) log into this user and then run the File 2 Script. This user essentially clears the environment before inserting new data into tables. Insert queries are shown only for limited test cases. Exceptional handling will be implemented when final procedures are created to update tables.

Note: No other user has access to create or drop tables, sequences, procedures etc. If you use another user tries you would get an output asking the user to reach out to admin